# Final Project Presentation

Nomor Kelompok: 5
Nama Mentor: Ramdhan Hidayat
Nama:

- <Juwita Natalia Sinaga>
- <Rima Chusnul Magfiroh>

Machine Learning Class

**Program Studi Independen Bersertifikat**
**Zenius Bersama Kampus Merdeka**

# Petunjuk

- Waktu presentasi adalah 5 menit (tentatif, tergantung dari banyaknya kelompok yang mendaftarkan diri)
- Waktu tanya jawab adalah 5 menit
- Silakan menambahkan gambar/visualisasi pada slide presentasi
- Upayakan agar tetap dalam format poin-poin (ingat, ini presentasi, bukan esai)
- Jangan masukkan *code* ke dalam slide presentasi (tidak usah memasukan screenshot jupyter notebook)

# Latar Belakang

# Latar Belakang Project

Sumber Data:
https://www.kaggle.com/datasets/yasserh/loan-default-dataset

Problem: **classification**

Tujuan:

- memprediksi status loan berdasarkan faktor-faktor yang diperhatikan perbankan agar mereka tidak salah dalam memberikan *loan* kepada nasabah.

# Explorasi Data dan Visualisasi

# Business Understanding

Menganalisis dataset *loan* (pinjaman perbankan) yang menyimpan data historis nasabah bank yang cenderung default (gagal bayar pinjaman) atau tidak. Mengidentifikasi nasabah yang berisiko tinggi untuk gagal bayar adalah salah satu cara untuk meminimalisir kerugian pemberi pinjaman. Untuk itu, kita akan coba memprediksi kemungkinan nasabah gagal bayar menggunakan prediktor-prediktor yang disediakan. Target kolomnya adalah ‘Status’ dengan keterangan 0 ditolak dan 1 diterima.

# Data Cleansing

Dataset Loan default memiliki 34 kolom dan 148670 baris. Dengan kolomnya adalah

```
df.columns
```

```
Index(['ID', 'year', 'loan_limit', 'Gender', 'approv_in_adv', 'loan_type',
       'loan_purpose', 'Credit_Worthiness', 'open_credit',
       'business_or_commercial', 'loan_amount', 'rate_of_interest',
       'Interest_rate_spread', 'Upfront_charges', 'term', 'Neg_ammortization',
       'interest_only', 'lump_sum_payment', 'property_value',
       'construction_type', 'occupancy_type', 'Secured_by', 'total_units',
       'income', 'credit_type', 'Credit_Score', 'co-applicant_credit_type',
       'age', 'submission_of_application', 'LTV', 'Region', 'Security_Type',
       'Status', 'dtir1'],
      dtype='object')
```

# Data Cleansing

Terdapat kejanggalan pada tipe data kolom `total_units` dan `age` yang memiliki tipe data object, padahal total unit dan usia seharusnya bertipe data integer. Kolom `total_units` dikelompokkan berdasarkan jumlah unitnya dan kolom `age` dikelompokkan berdasarkan range usia tertentu sehingga kedua kolom ini memiliki tipe data object. Nilai-nilai pada `total_units` akan diubah menjadi integer.

Kolom `Gender` memiliki nilai 'Sex Not Available' yang sama dengan nilai NaN, maka dari itu nilai 'Sex Not Available akan diganti dengan NaN'

# Data Cleansing

Dataset df memiliki banyak kolom yang memiliki nilai kosong (missing value), terutama di kolom Gender, rate_of_interest, Interest_rate_spread, dan juga Upfront_charges. Akan dilihat persebaran missing value ini berdasarkan Statusnya.

Kolom rate_of_interest, Interest_rate_spread, dan Upfront_charges akan dihapus karena nilai kosong pada Status 1-nya sangat banyak.

```
df.Gender.isnull().groupby([df['Status']]).sum().astype(int)
```

```
Status
0    26892
1    10767
Name: Gender, dtype: int32
```

```
df.rate_of_interest.isnull().groupby([df['Status']]).sum().astype(int)
```

```
Status
0        0
1    36439
Name: rate_of_interest, dtype: int32
```

```
df.Interest_rate_spread.isnull().groupby([df['Status']]).sum().astype(int)
```

```
Status
0        0
1    36639
Name: Interest_rate_spread, dtype: int32
```

```
df.Upfront_charges.isnull().groupby([df['Status']]).sum().astype(int)
```

```
Status
0     3156
1    36486
Name: Upfront_charges, dtype: int32
```

# Data Cleansing

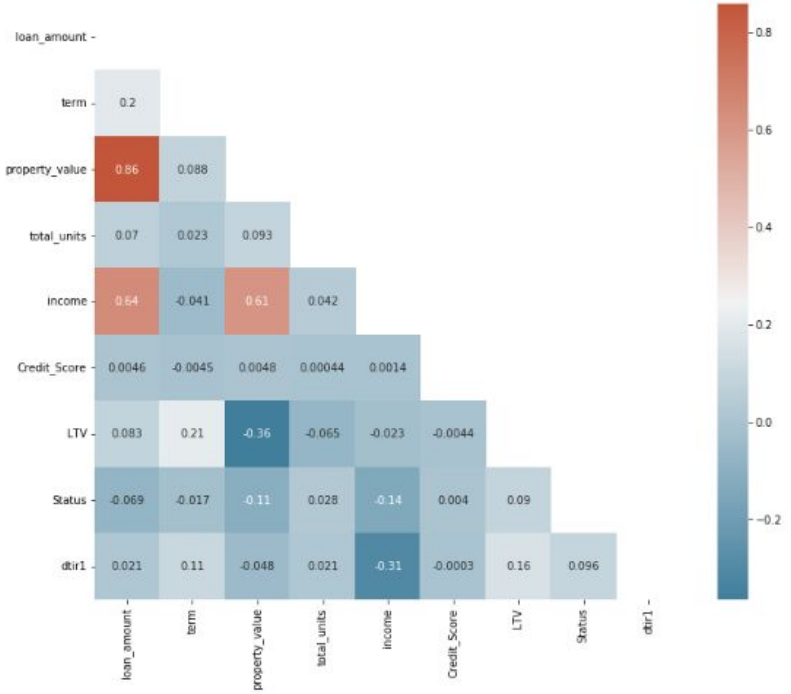
Kolom ID tidak berpengaruh terhadap Status, begitu pula dengan kolom year karena hanya memiliki satu nilai yaitu 2019. Maka dari itu, kolom ID dan year akan dihapus.

```
df.drop(['ID','year'], axis = 'columns', inplace = True)
```

# Exploratory Data Analysis

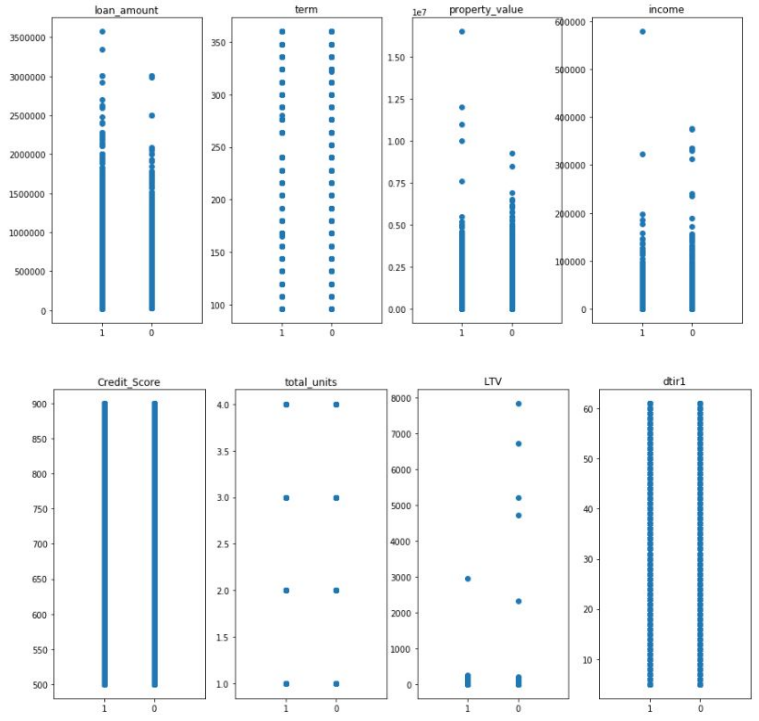
Berikut merupakan heatmap yang menunjukkan korelasi antar tiap kolom numerik.

Dari heatmap di samping, bisa kita simpulkan bahwa property_value dan loan_amount memiliki korelasi yang tinggi. Sehingga jika salah satu dipilih menjadi prediktor, maka yang lainnya harus diserakan. Begitu pula kolom income dengan loan_amount dan kolom income dengan property_value memiliki korelasi yang cukup tinggi. Semakin tinggi property value-nya, semakin tinggi income dan juga loan amountnya, begitu pula sebaliknya.



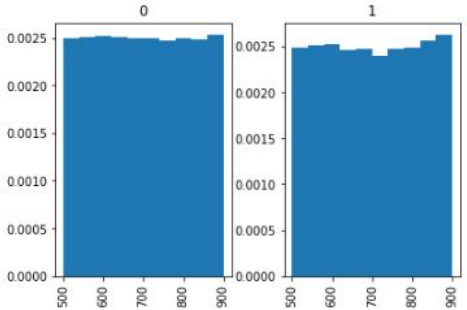
# Exploratory Data Analysis

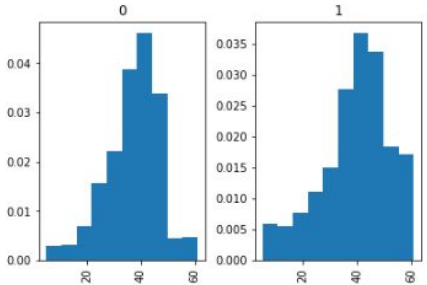
Berikutnya akan dilihat hubungan antar kolom numerik dengan kolom Status dengan menggunakan scatterplot, Ternyata pinjaman yang statusnya diterima jumlah maksimalnya justru lebih tinggi dibandingkan dengan yang statusnya ditolak, begitu pula dengan jumlah minimalnya lebih rendah. Sehingga dapat disimpulkan loan_amount tidak memengaruhi diterima/ditolaknya pinjaman, karena loan_amount yang tinggi juga mengindikasikan bahwa si peminjam punya income yang lebih tinggi, begitu pula sebaliknya. Karena property_value dan income memiliki korelasi yang cukup tinggi dengan loan_amount, kita tidak akan menggunakan kolom property_value dan income sebagai prediktor.



# Exploratory Data Analysis

Kolom term dan LTV juga tidak begitu berpengaruh terhadap Status sehingga kolom tersebut tidak akan dipakai sebagai prediktor. Terakhir, kolom Credit_Score dan dtir1 memiliki rata-rata sedikit lebih tinggi pada pinjaman yang diterima. Berikut ditampilkan histogram Credit_Score dan dtir1 berdasarkan statusnya.

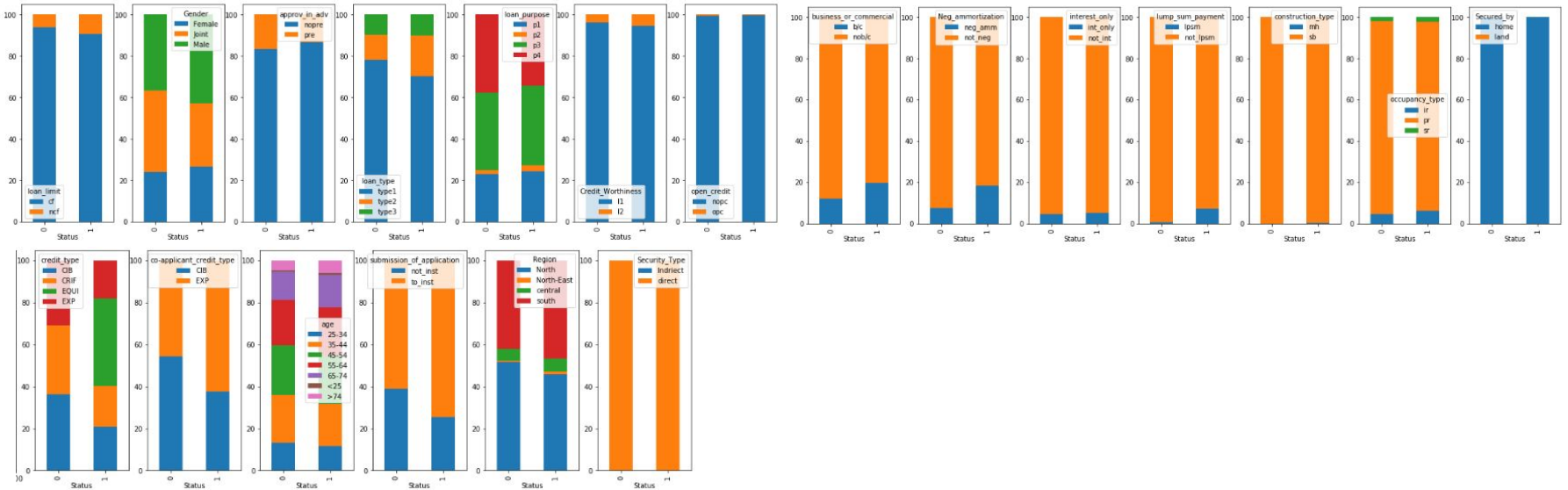



Pada histogram di samping dapat dilihat bahwa pada histogram memiliki perbedaan yang cukup signifikan anatar 0 dan 1, sehingga Credit_Score dan dtir1 akan digunakan.

# Exploratory Data Analysis

Selanjutnya akan dipilih kolom-kolom categorical yang akan menjadi feature dengan menggunakan barplot.



# Exploratory Data Analysis

Pada barplot di atas dapat dilihat dengan jelas bahwa kolom open_credit, interest_only, construction_type, occupancy_type, Secured_by, credit_type, co-applicant_credit_type, region, dan Security_Type tidak memiliki perbedaan jumlah yang signifikan antara pinjaman yang diterima dan tidak. Maka dari itu, kolom-kolom tersebut akan dihapus dan tidak digunakan sebagai prediktor.

Kolom-kolom tersebut akan digunakan sebagai prediktor adalah loan_type, loan_amount, property_value, income, Credit_Score, age, term, dtir1

# Modelling

# One Hot Encoding

Sebelum melakukan one-hot encoding, akan dilakukan resampling data terlebih dahulu karena terdapat perbedaan banyaknya data yang disetujui dan ditolak pada kolom Status. Dengan menggunakan downsampled.

```
from sklearn.utils import resample

#create two different dataframe of majority and minority class
df_majority = df_update[(df_update['Status']==0)]
df_minority = df_update[(df_update['Status']==1)]
# upsample minority class
df_majority_downsampled = resample(df_majority,
                                   replace=True,    # sample with replacement
                                   n_samples=36639, # to match majority class
                                   random_state=42)  # reproducible results
# Combine majority class with upsampled minority class
df_downsampled = pd.concat([df_majority_downsampled, df_minority])
```

```
df_downsampled['Status'].value_counts()
```

```
1    36639
0    36639
Name: Status, dtype: int64
```

# Random Forest

```
X = df_final.loc[:, df_final.columns != 'Status']
y = df_final["Status"]
```

```
# splitting X and y into training and testing sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.4,
                                                    random_state=1)
```

```
from sklearn.ensemble import RandomForestClassifier

classifier_rf = RandomForestClassifier(random_state=42, n_jobs=-1, max_depth=5,
                                       n_estimators=100, oob_score=True)

classifier_rf.fit(X_train, y_train)
```

# Hyperparameter Tuning

```
#Evaluasi model menggunakan AUC
from sklearn.metrics import roc_curve, auc
#fpr, tpr, thresholds = roc_curve(y_test, y_rf, pos_label=1) # pos_label: positive label
#print(auc(fpr, tpr))
fpr, tpr, thresholds = roc_curve(y_test, y_classifier_rf, pos_label=1) # pos_label: positive label
print(auc(fpr, tpr))
```

# Conclusion

- Classification model yang kami gunakan pada model ini menggunakan metode random forest dengan akurasi 76%. Kami melakukan hyperparameter tuning untuk melakukan improvement terhadap model awal, diperoleh akurasi model akhir adalah %. Masalah imbalanced problem sudah kami atasi dengan resample dataset sehingga data yang digunakan untuk train dan test model seimbang.
- Feature-feature penting yang memengaruhi disetujui atau tidaknya suatu permohonan pinjaman adalah loan_type, loan_amount, property_value, income, Credit_Score, age, term, dtir1

`loan_type` dan `loan_amount` berpengaruh dimana loan_type type 2 lebih banyak disetujui. Peminjam pada usia>55 tahun lebih banyak diterima dibandingkan dengan peminjam pada usia<55 tahun.

Perusahaan sebaiknya memperhatikan tipe loan yang diberikan. Jika tipe loan memengaruhi default atau tidaknya suatu pinjaman, maka seharusnya terdapat kriteria siapa saja yang dapat memilih tipe loan tersebut.

# Terima kasih!

## Ada pertanyaan?